



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/885,077	06/21/2001	Kevin Wade Jameson		3816
29684	7590	06/03/2004	EXAMINER	
KEVIN JAMESON 148 EDGE BANK PLACE NW CALGARY, AB T3A 4L4 CANADA			ROCHE, TRENTON J	
		ART UNIT	PAPER NUMBER	5
		2124		
DATE MAILED: 06/03/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>
	09/885,077	JAMESON, KEVIN WADE
	<b>Examiner</b>	<b>Art Unit</b>
	Trent J Roche	2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) Responsive to communication(s) filed on 23 July 2002.  
 2a) This action is FINAL.                    2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) Claim(s) 1-27 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 1-27 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on 21 June 2001 is/are: a) accepted or b) objected to by the Examiner.  
     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |                                                                                                                          |                                                                             |
|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                              | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                     | Paper No(s)/Mail Date. _____ .                                              |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ . | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
|                                                                                                                          | 6) <input type="checkbox"/> Other: _____ .                                  |

### **DETAILED ACTION**

1. This office action is responsive to Amendment A filed 23 July 2002.
2. Claims 1-27 have been examined.

#### *Information Disclosure Statement*

3. The information disclosure statement (IDS) submitted on 21 June 2001 stated that the applicant conducted a professional search but found no relevant prior art.

#### *Claim Rejections - 35 USC § 112*

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
5. Claims 1-27 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1-27 recite functional language stating that the claimed limitations provide solutions to a number of problems surrounding makefiles, and that their solutions provide human programmers the ability to perform actions in an automated, scalable way that was not previously available. However, it is unclear what steps of the claims are associated with the functional language. For example, in claim 1, a solution to the collection makefile generator problem is established, but it is unclear as to what step or process in the claim provides this solution, as opposed to alternate prior art which discloses the generation of makefiles. Furthermore, no indication is made that any of the steps provide an *automated and scalable* (emphasis added) way to generate collection makefiles. As

Art Unit: 2124

such, it is unclear as to what performs the claimed functionality. Claims 2-27 are indefinite for similar reasoning.

*Claim Rejections - 35 USC § 102*

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-4, 6, 10-13, 15, 19-22 and 24 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent 6,032,198 to Fujii et al, hereafter referred to as Fujii.

**Regarding claim 1:**

Fujii teaches:

- a collection makefile generator process for generating makefiles for collections (Note Figure 2 and the corresponding sections of the disclosure)
- receiving a request to generate a makefile for a collection (“The generation of the makefile is performed by the makefile generation unit which is one subprogram of the editor unit” in col. 8 lines 63-65. A request is inherently sent to the makefile generation unit, so that the unit may generate the makefile.)
- accessing collection information for said collection (“the makefile generation unit fetches the skeleton source and header files names and the stub source and header file names from the document management function in accordance with the object ID’s...” in col. 9 lines 20-23)

Art Unit: 2124

- generating a makefile for said collection (“the makefile generation unit generates a compile command for each source file name acquired...and writes the generated compile command into the makefile” in col. 9 lines 46-49)
- wherein collections are data structures comprised of a collection specifier (“an object ID” in col. 9 line 6)
- collection content containing zero or more collection content files (“the skeleton source and header files names and the stub source and header file names from the document management function in accordance with the object ID’s...” in col. 9 lines 21-23)
- wherein a collection specifier contains information about a collection instance, and wherein collection membership information describes collection content (“The makefile generation unit acquires interface definition information associated with the application symbol information from the interface information management unit... the skeleton source and header files names and the stub source and header file names from the document management function in accordance with the object ID’s...” in col. 9 lines 7-23)

substantially as claimed. Fujii does not explicitly describe providing a solution to the collection makefile generator problem, and thereby enabling human programmers to generate collection makefiles in a fully-automated, scalable way, however, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii. Accordingly, the system of Fujii must inherently provide a solution to the collection makefile generator problem, and thereby enable human programmers to generate collection makefiles in a fully-automated, scalable way.

**Regarding claim 2:**

Art Unit: 2124

The rejection of claim 1 is incorporated, and further, Fujii discloses using makefile service and makefile fragment template information as claimed (“the makefile generation unit fetches the skeleton source and header files names and the stub source and header file names from the document management function in accordance with the object ID’s...” in col. 9 lines 20-23. The skeleton code is used as a template.) Fujii does not explicitly describe providing a solution to the makefile customization problem, and thereby enabling human programmers to extensively customize the makefile generation process by controlling the information content of fragments used to implement makefile services in an automated, scalable way, however, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii. Accordingly, the system of Fujii must inherently provide a solution to the makefile customization problem, and thereby enable human programmers to extensively customize the makefile generation process by controlling the information content of fragments used to implement makefile services in an automated, scalable way.

**Regarding claim 3:**

The rejection of claim 1 is incorporated, and further, Fujii discloses using substitution strings representing collection instance values to replace placeholder strings in makefile fragments as claimed (“the makefile generation unit fetches the skeleton source and header files names and the stub source and header file names from the document management function in accordance with the object ID’s...” in col. 9 lines 20-23. The skeleton source is used as placeholders until the stub source replaces the generic information with detailed information.) Fujii does not explicitly describe providing a solution to the multiple product naming problem, nor helping to solve the makefile customization program, and thereby enabling human programmers to use practical placeholder

Art Unit: 2124

strings to represent useful collection instance information values in makefile fragments in a more convenient way, however, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii. Accordingly, the system of Fujii must inherently provide a solution to the multiple product naming problem and help to solve the makefile customization problem, and thereby enable human programmers to use practical placeholder strings to represent useful collection instance information values in makefile fragments in a more convenient way.

**Regarding claim 4:**

The rejection of claim 1 is incorporated, and further, Fujii discloses using standalone makefile service information as claimed (“Application symbol information of all the thus specified operation providing applications is acquired from the application information management unit...” in col. 9 lines 32-35) Fujii does not explicitly describe helping to solve the makefile customization program, and thereby enabling human programmers to further additively customize generated makefiles by specifying ad hoc additions to the generated makefiles in the form of standalone makefile service statements within collection specifiers, however, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii. Accordingly, the system of Fujii must inherently help to solve the makefile customization problem, and thereby enable human programmers to further additively customize generated makefiles by specifying ad hoc additions to the generated makefiles in the form of standalone makefile service statements within collection specifiers.

**Regarding claim 6:**

Art Unit: 2124

The rejection of claim 1 is incorporated, and further, Fujii discloses using information selected from the group consisting of include file search directory information and library file search directory information as claimed (“the makefile generation unit fetches the skeleton source and header files names and the stub source and header file names from the document management function in accordance with the object ID’s stored in interface definition information...” in col. 9 lines 20-24) Fujii does not explicitly describe providing solutions to the include file directory problem and the library file directory problem, and thereby helping to improve the runtime performance of compilers and linkers by providing them with optimal lists of include file and library search directories to search during compilation and linking operations, however, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii. Accordingly, the system of Fujii must inherently provide solutions to the include file directory problem and the library file directory problem, and thereby help to improve the runtime performance of compilers and linkers by providing them with optimal lists of include file and library search directories to search during compilation and linking operations.

**Regarding claims 10-13, 15, 19-22 and 24:**

Claims 10-13 and claims 15 and 19-22 and 24 recite a programmable device and a computer readable memory encoded with data, respectively, for directing a computer to perform the process of claims 1-4 and 6, respectively, and are therefore rejected for the reasons set forth in connection with claims 1-4 and 6, respectively.

*Claim Rejections - 35 USC § 103*

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 5, 7-9, 14, 16-18, 23 and 25-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,032,198 to Fujii et al, hereafter referred to as Fujii, in view of U.S. Patent Publication 2002/0147855 A1 to Lu.

**Regarding claim 5:**

The rejection of claim 1 is incorporated, and further, Fujii does not explicitly disclose using virtual platform information. Lu discloses in an analogous system for generating makefiles the use of virtual platform information as claimed (“preferably written in a platform independent language such as, for example, Java...” in paragraph 0064). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the platform independence of Lu with the system for generating makefiles disclosed by Fujii, as this would allow a developer to create a single version of a makefile which is independent of the shell and makefile interpreters employed, as stated in paragraph 0147 of Lu. Neither Fujii nor Lu explicitly describe providing a solution to the makefile template sharing problem, and thereby enabling human programmers to minimize makefile fragment creation and maintenance costs by sharing makefile fragments across collection, product, file, action, and platform types, however, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii modified by Lu. Accordingly, the system of Fujii modified by

Lu must inherently provide a solution to the makefile template sharing problem, and thereby enable human programmers to minimize makefile fragment creation and maintenance costs by sharing makefile fragments across collection, product, file, action, and platform types.

**Regarding claim 7:**

The rejection of claim 1 is incorporated, and further, Fujii does not explicitly disclose a product build order. Lu discloses in an analogous system for generating makefiles a product build order as claimed (“another make utility will be launched as a new make process which will operate to block further interpretation of makefile wrapper until make utility completes its operations” in paragraph 0134. This prevents the make command from operating out-of-order and causing errors.). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the product build order of Lu with the system for generating makefiles disclosed by Fujii, as this would ensure that a single make operation can be completed without another operation interrupting the process. Neither Fujii nor Lu explicitly describe providing a solution to the multiple product build order problem, and thereby enabling human programmers to specify the construction of multiple output products from a single collection, even where said multiple products must be processed in a particular build order in order to ensure correct results, however, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii modified by Lu. Accordingly, the system of Fujii modified by Lu must inherently provide a solution to the multiple product build order problem, and thereby enable human programmers to specify the construction of multiple output products from a single collection, even where said multiple products must be processed in a particular build order in order to ensure correct results.

**Regarding claim 8:**

The rejection of claim 1 is incorporated, and further, Fujii does not explicitly disclose a file build order. Lu discloses in an analogous system for generating makefiles a file build order as claimed (“and the order in which these targets should be built” in paragraph 0026). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the file build order of Lu with the system for generating makefiles disclosed by Fujii, as this would ensure that the files being processed would be read in the correct order, ensuring that the product being built would be fully operational and error free. Neither Fujii nor Lu explicitly describe providing a solution to the product file build order problem, and thereby enabling human programmers to specify a proper build order for the multiple product file types within a single collection product, even where the multiple product files must be processed in a particular build order in order to ensure correct results, however, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii modified by Lu. Accordingly, the system of Fujii modified by Lu must inherently provide a solution to the product file build order problem, and thereby enable human programmers to specify a proper build order for the multiple product file types within a single collection product, even where the multiple product files must be processed in a particular build order in order to ensure correct results

**Regarding claim 9:**

The rejection of claim 1 is incorporated, and further, Fujii does not explicitly parallelism limit information to calculate and generate parallel makefile targets. Lu discloses in an analogous system for generating makefiles parallelism limit information to calculate and generate parallel makefile targets as claimed (“provides the identity of targets for parallelization and the order in which these

targets should be built" in paragraph 0026). It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the parallelism limit information of Lu with the system for generating makefiles disclosed by Fujii, as this would enable some instructions to be executed in parallel, thus providing reduced compiling time and improving productivity, as stated by Lu in paragraph 0026. Lu further discloses increasing the productivity of human programmers by generating makefiles that can be executed in parallel to reduce the time required to perform makefile operations (Note paragraph 0026), however, neither Fujii nor Lu explicitly describe providing a solution to the makefile parallel processing problem. However, the structural elements described in the claim have been shown to exist in the system disclosed by Fujii modified by Lu. Accordingly, the system of Fujii modified by Lu must inherently provide a solution to the makefile parallel processing problem.

**Regarding claims 14, 16-18, 23 and 25-27:**

Claims 14 and 16-18 and claims 23 and 25-27 recite a programmable device and a computer readable memory encoded with data, respectively, for directing a computer to perform the process of claims 5 and 7-9, respectively, and are therefore rejected for the reasons set forth in connection with claims 5 and 7-9, respectively.

***Conclusion***

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Trent J Roche whose telephone number is (703)305-4627. The examiner can normally be reached on Monday - Friday, 9:00 am - 6:30 pm.

Art Unit: 2124

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Trent J Roche  
Examiner  
Art Unit 2124

TJR

*Kakali Chaki*  
KAKALI CHAKI  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100